Received: 26 March 2019 | Revised: 4 September 2019 | Accepted: 16 October 2019
DOI: 10.1002/cpe.5604

WILEY

RESEARCH ARTICLE

An improved term weighting scheme for text classification

Zhong Tang | Wenqiang Li | Yan Li

School of Mechanical Engineering, Sichuan University, Sichuan Province's Key Laboratory of Innovation Methodology and Creative Design, Chengdu, China

Correspondence
Wenqiang Li, School of Mechanical Engineering, Sichuan University, No. 24 South Section 1, Yihuan Road, Chengdu 610065, China.
Email: liwenqiang@scu.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: 51435011; Science & Technology Ministry Innovation Method Program, Grant/Award Number: 2017IM040100; Sichuan Applied Foundation Project, Grant/Award Number: 2018JY0119

Summary

Text representation is a necessary and primary procedure in performing text classification (TC), which first needs to be obtained through an information-rich term weighting scheme to achieve higher TC performance. So far, term frequency-inverse document frequency (TF-IDF) is the most widely used term weighting scheme, but it suffers from two deficiencies. First, the global weighting factors IDF in TF-IDF approaches infinity if a certain term does not occur in a text. Second, the IDF is equal to zero if a certain term appears in any text. To offset these drawbacks, we first conduct an in-depth analysis of the current term weighting schemes, and subsequently, an improved term weighting scheme called term frequency-inverse exponential frequency (TF-IEF) and its various variants are proposed. The proposed method replaces IDF with the new global weighting factor IEF to characterize the global weighting factor log-like IDF in the corpus, which can greatly reduce the effect of feature (term) with high local weighting factor TF in term weighting. As a result, a more representative feature can be generated. We carried out a series of experiments on two commonly used data sets (corpora) utilizing Naïve Bayes and support vector machine classifiers to validate the performance of our proposed schemes. Experimental results explicitly reveal that the proposed term weighting schemes come with better performance than the compared schemes.

KEYWORDS
feature selection, term weighting, text classification, text representation, TF-IEF

1 | INTRODUCTION

With the rapid development of Internet technology, the volume of digital texts, web pages, messages, and so on, all of which available online are growing exponentially. As a consequence, effective organization and process for the huge amount of textual data are becoming increasingly important. Text classification (TC), also called text categorization, becomes the key technology to achieve the above purposes.[1,2] Recently, many works have been proposed for the TC. Canuto et al[3] presented a use of a multiobjective optimization strategy to reduce the number of metafeatures while maximizing the classification effectiveness. Do and Poulet[4] proposed a novel fast and accurate parallel local support vector machine (SVM) algorithm for classifying very high-dimensional input spaces and large-scale multiclass data sets. Gao et al[5] developed a structured sparse representation classifier to short TC. Uysal and Gunal[6] extensively examined the influence of preprocessing on TC in terms of various aspects such as text language, text domain, and classification accuracy. Ay Karakuş et al[7] introduced a binary classification study for the Turkish language on movie reviews. In addition, text categorization has multiple applications, such as web page classification,[8,9] spam detection,[10,11] author identification,[12,13] and customer relationship management.[14]

TC is such a task to categorize unlabelled natural language texts into a predefined set of thematic classes based on their content.[15] Therefore, a TC task starts with a training set $D = \{d_1, d_2, \ldots, d_k\}$ of documents that are already labeled with a class $C = \{c_1, c_2, \ldots, c_i\}$, where d_k is the kth document and c_i is the ith category.[16] Before applying classifiers, texts first need to be transformed into numerical vectors in an appropriate way so that the classifier can understand and relate them, and this stage is named text representation and is essential for TC tasks.[17,18] At present, the vector space model (VSM) is the most commonly utilized and effective technologies for text representation in which each text is represented as a feature vector whose components are the term weights.[19] In this type of model, features can be of different types, including terms (or words), syntactic, phrases, or any other indexing units.[20,21] Among these indexing units, term is the smallest constituents of text and plays a critical role

in the TC task, so the frequency of occurrence of these terms (words or features) in the texts is represented as a vector and then forms a vector space of all the features.[22] Moreover, a text is considered a bag of words (BOW), mapped into a feature vector.[23] For example, in the VSM, a text is regarded as a vector of weighted features (or terms) such that $d_k = (t_1, t_2, \ldots, t_n)$, and a corresponding weight vector $w = (w_1, w_2, \ldots, w_n)$, where n is the number of chosen features (terms), $w_1, w_2, \ldots, w_n$ are the weights of $t_1, t_2, \ldots, t_n$, respectively, based on the employed term weighting scheme.[21]

It is true that text is one of the most efficient means of information preservation and dissemination,[17] which may contain hundreds or even thousands of terms (features). To organize and categorize a large number of texts more efficiently, text representation is a necessary and primary procedure that enables the classifier algorithms can handle the textual contents.[20] We must note, however, that the term is only the carrier of information, rather than itself. So, if we do not use terms, can other carriers (such as numbers) represent equally meaningful information? There is no doubt that the answer is obviously yes. Therefore, no matter which carrier utilized to identify the contents of the texts, each term (feature) first must be assigned an appropriate value (ie, weight), it measures the importance of this term and reflects how much its contributes to the TC task.[20]

So far, existing methods for text representation are mainly from two perspectives: semantic-based term weight methods and statistic-based term weight methods. The semantic method concentrates not only on meaning of the words but also on hidden semantic relationships between words and consequently between documents.[24] Rao et al[25] presented a novel neural network model with two hidden layers to learn continuous text representation, which can capture semantic information between words and sentences. Nguyen et al[26] proposed a new approach for determining the degree of semantic similarity between pairs of short texts by utilizing word embeddings. Kastrati et al[27] introduced a semantically rich document representation model for automatically classifying texts using deep learning. In order to consider the semantic connections between words, Zhang et al[28] proposed an approach for sentiment classification based on Word2Vec and SVM^{perf}. Ay Karakuş et al[7] presented a sentiment classification study for the Turkish movie reviews using Word2Vec model and two deep learning networks (ie, convolutional neural networks and long short-term memory). By the way, Word2Vec can capture the semantic information of words based on the distributed hypothesis,[29] which states that words from the similar contexts will have similar meanings.[30] According to the distributed hypothesis, Word2Vec embeds words into continuous vector space by using neural network language models.[31] Additionaly, in order to make full use of the word sequence information to improve the performance of TC, Doc2Vec method based on Word2Vec is developed recently. Doc2Vec extends Word2Vec from the word level to the document level.[32,33] Kim et al[31] developed the bag-of-concepts approach for representing document vectors, which combines the advantages of the BOW method and Doc2Vec to overcome their limitations. In order to increase the variety of feature sets for TC, Kim et al[33] transformed a text using three text representation schemes, ie, term frequency-inverse document frequency (TF-IDF), latent Dirichlet allocation, and Doc2Vec.

Nevertheless, semantic-based term weight methods cannot significantly improve classification performance in addition to being more complex than statistical counterparts. Therefore, the term weighting scheme based on statistics is still the main research direction for TC tasks.[2] Lan et al[20] presented a novel supervised term weighting scheme (named term frequency-relevance frequency or TF-RF) to improve the terms' discriminating power for TC tasks. Maisonnave et al[34] developed a supervised term weighting scheme based on the notions of descriptive and discriminative relevance. Sinoara et al[35] proposed two methods to represent document collections based on embedded representations of both words and word senses. Zheng et al[36] built a flexible and easily extensible framework for term representation to unify various Twitter-specific features. Sabbah et al[9] developed a hybridized approach that utilizes the use of a wider range of term weighting schemes simultaneously such as TF, DF, TF-IDF, entropy, and so on. Since then, four modified frequency-based term weighting schemes that consider missing terms when computing the weights of existing terms are proposed.[21] Salton et al[37] claimed that normalized TF-IDF is the best document weighting function by investigating many term weighting methods in the information retrieval domain. Zhang et al[16] has comparatively studied the performances of three text representation schemes: TF-IDF, latent semantic indexing (LSI), and multiword in TC. This outcome has shown that TF-IDF has better statistical quality than the LSI and multiword methods. Nevertheless, Chen et al[2] stated that the famous TF-IDF term weighting scheme is not fully effective for TC purpose, so a novel term weighting method, called term frequency-inverse gravity moment (TF-IGM) is presented. It may be noted, however, that TF-IGM scheme assigns equal scores (ie, same weights) to terms for some extreme cases. To offset this shortcoming, two new term weighting schemes based on inverse gravity moment are proposed.[38]

Undeniably, all the aforementioned schemes make a significant contribution in text representation. We must note, however, that different representations of the same text may incorporate different feature information.[39] Therefore, how to assign appropriate weights to terms in some way is a primary issue for TC, which will directly influence the classification performance. At present, TF-IDF is the most widely used term weighting scheme, but the global weighting factor IDF in TF-IDF will appear two extremes, that is, approaches infinity and equal to zero. To further explore these problems, an improved term weighting scheme named term frequency-inverse exponential frequency (TF-IEF) and its various variants are proposed in this study. The motivation behind the proposed scheme came from the thought that text representation first needs to be obtained through an information-rich term weighting scheme. Furthermore, the comparison is conducted for different data sets and classification algorithms to evaluate the effectiveness of the proposed method. Experimental results show that the proposed schemes comparatively outperforms other existing baseline schemes.

The remainder of our paper is arranged as follows. Section 2 gives an overview of the existing term weighting scheme. Section 3 introduces the text representation problem using the TF-IDF weighting scheme, and then we depict the details of our weighting scheme. Section 4 briefly describes the experimental settings including the experimental data sets, the preprocessing, the feature selection methods, the classification algorithms, and the success measure used. Details of the experimental analysis and the related results are presented in Section 5. Section 6 concludes this paper.

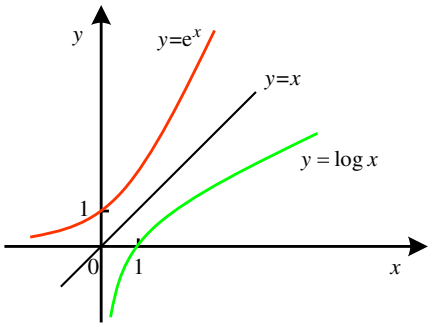

FIGURE 1 The logarithmic function and its inverse function curve

2 | ANALYSIS OF CURRENT TEXT REPRESENTATION TECHNIQUES FOR TC

For text representation, the term weighting schemes such as term frequency (TF), document frequency (DF), inverse DF (IDF), and TF-IDF are commonly used. In the following, three famous term weighting methods (ie, TF, IDF, and TF-IDF) for TC are analyzed.

2.1 | Term frequency

The TF representation is one of the most popular and simplest term weighting schemes, because the raw TF in the document is adopted. The TF weighting scheme supports the assumption that more frequent term in the text indicates a more important term. It is obvious that the TF is a local weighting scheme since it depends on the number of times a certain term appears in a text. To eliminate the effect of document length, the normalized TF is utilized to limit the term weight falls in [0, 1], so the L_2-normalization is applied to normalize the term weight in this research,[40] defined by Equation (1).

$$TF\left(t_j, d_k\right) = \frac{tf\left(t_j, d_k\right)}{\sqrt{\sum_{j=1}^{n} tf\left(t_j, d_k\right)^2}}, \tag{1}$$

where n is the number of chosen features (terms) and $tf(t_j, d_k)$ is the TF of feature t_j in document d_k. In Equation (1), the numerator is the raw frequency of the term t_j (denoted by $tf(t_j, d_k)$) in document d_k, and in fact, the denominator is the Euclidean norm of the document d_k (denoted by $\sqrt{\sum_{j=1}^{n} tf(t_j, d_k)^2}$). Note that the $tf(t_j, d_k)$ has many variants such as $\log(tf(t_j, d_k))$, $\log(tf(t_j, d_k)+1)$, and $\log(tf(t_j, d_k))+1$.

2.2 | Inverse document frequency

TF only count whether a certain term occurs within a text, so TF alone may not have enough the distinction capacity to pick up all relevant texts from other irrelevant texts. To solve this problem, an IDF that concerned with the collection distribution has been introduced to enhance the term's discriminative capacity for TC tasks.[20] IDF is evolved from DF, and it represents the assumption that the terms appear in more different documents is regarded to be less important and vice versa.[21]

Accordingly, let the DF $df(t_j)$ be the number of the document in which term t_j appears at least once. With respect to term t_j, its IDF $IDF(t_j)$ can be expressed as Equation (2),[37] where N denote the total number of documents in the training set.

$$IDF\left(t_j\right) = \log\left(N/df\left(t_j\right)\right) \tag{2}$$

Unlike the TF scheme, IDF is a global term weighting scheme, which denotes the relationship between terms and documents in the collection. More importantly, the IDF depends on the logarithmic function (ie, log$x$) where the base is usually e (natural number) in Equation (2), it is obviously a monotonically increasing function, as shown in Figure 1. Actually, the $IDF(t_j)$ gradually increases as the DF $df(t_j)$ decreases, and vice versa. Furthermore, the $IDF(t_j)$ also has a number of variants such as $\log(N/(df(t_j)+1))$, $\log(N/df(t_j)+1)$, and $\log(N/df(t_j))+1$.

2.3 | Term frequency-inverse document frequency

Currently, TF-IDF is the most widely used term weighting scheme for TC task.[41] It is based on the assumption that the term occur less frequent in the corpus is the more important term (feature) in the text.[21] Since TF-IDF is extended from IDF, which is first presented by Jones,[42] it is also a global statistical measure. TF-IDF of a term t_j in document d_k(denoted as $TFIDF(t_j, d_k)$) is the product of the TF (denoted as $tf(t_j, d_k)$) and the IDF (denoted as $IDF(t_j)$), then the classical definition of TF-IDF used for term weighting obtained by

$$TFIDF\left(t_j, d_k\right) = tf\left(t_j, d_k\right) \cdot IDF\left(t_j\right) \tag{3}$$

In this study, we utilizes the following TF-IDF formula to compute the weight of term t_j in the document d_k, which is also performed the normalize to eliminate the length effect. Hence, the formula for TF-IDF can be reformulated as follows:

$$TFIDF\left(t_j, d_k\right) = \frac{tf\left(t_j, d_k\right) \cdot \log\left(N/df\left(t_j\right)\right)}{\sqrt{\sum_{j=1}^{n}\left(tf\left(t_j, d_k\right) \cdot \log\left(N/df\left(t_j\right)\right)\right)^2}} \tag{4}$$

3 | IMPROVED WEIGHTING SCHEME AND ITS VARIOUS VARIANTS

In this section, we first describe the text representation problem using the TF-IDF weighting scheme, followed by the ten weighting schemes combined with TF are summarized. According to these previous studies, an improved term weighting scheme named TF-IEF as well as its various variants are presented for TC.

3.1 | Problem statement

As in the TF-IDF formula (see Equation (4)), the IDF values are obtained through the logarithmic function, so it naturally has some drawbacks. In particular, there are two major deficiencies of this scheme. First, if the term t_j does not occur in any document, ie, $df(t_j) = 0$, then the denominator of the proper number in the log probability ratio $\log(N/df(t_j))$ is equal to 0, which means that $IDF(t_j)$ no longer has any meaning in term weighting. Therefore, the $IDF(t_j)$ may yield a nonconvergence problem, since its will approaches infinity (ie, $IDF(t_j) \to +\infty$) at mathematics when $df(t_j)$ is near zero. In order to avoid division by zero, an alternative scheme is adopted recently, namely $\log(N/(df(t_j)+\varepsilon))$, where ε is an adjustable parameter to make the denominator larger than zero and is empirically determined. So, if $df(t_j)$ is smaller than ε, or even equal to 0, then the main influence on global weighting factor $IDF(t_j)$ is ε, rather than $df(t_j)$. Second, if the term t_j appears in any document, ie, $df(t_j) = N$, then the $IDF(t_j)$ score will be $\log(N/N) = 0$. This means that the weight $TFIDF(t_j, d_k)$ of term is always equal to 0. Similarly, some alternative schemes are adopted to handle the above issue, such as $\log(N/df(t_j)+\eta)$ (or $\log(N/df(t_j))+\eta$). However, we must also note that the main influence on global weighting factor $IDF(t_j)$ may be η, rather than $df(t_j)$, when $N/df(t_j)$ is less than η. From the above arguments, we claim that the global weighting factor $IDF(t_j)$ will appear two extremes (or deficiencies), namely, approaches infinity and equal to zero.

To obtain a more representative feature (term), researchers adopted the values of feature selections or other metrics to substitute IDF factor in TF-IDF[18]; they can be regarded as an adjustment coefficient for the TF-based local weighting factor. Table 1 shows the main term weighting schemes based on different feature selection approaches and other metrics. It is now apparent from Table 1 that the term weight is usually composed of two parts, which correspond respectively to the local weighting factor in a text and the global weighting factor in the data set. In particular, the weight of a term is always the product of TF and a certain global function in which TF is obviously utilized as the local weighting factor among these weighting schemes. Because in any case, counting the number of times of each term appears in a text is a necessary procedure for TC tasks, which suggests that these weighting methods are based on the BOW model.

3.2 | Inverse exponential frequency

Based on the above facts, we identify that the main difference between different term weighting methods is how to compute the global weighting factor, which increases the term's discriminating power in term weighting. In consideration of the drawbacks of IDF in TF-IDF, we retain the TF of the term t_j in our scheme. Now, let us consider the global weighting of the term t_j. As we mentioned before, different term weighting schemes are derived from different ideas (assumptions) for terms' characteristics in texts. From Equation (4), we can easily find that TF-IDF

TABLE 1 Summary of the weighting schemes based on different feature selection methods and other metrics

Weighting scheme	Name	Description	Reference
TF. Inverse document frequency	TF-IDF	Multiply TF by an inverse document frequency function	Jones[42] and Salton et al[41]
TF. Chi-square	TF-CHI2	Multiply TF by a chi-squared statistic function	Debole et al[43]
TF. Information gain	TF-IG	Multiply TF by an information gain function	Debole et al[43] and Lan et al[20]
TF. Mutual information	TF-MI	Multiply TF by a mutual information function	Altınçay et al[44] and Ren et al[45]
TF. Odds ratio	TF-OR	Multiply TF by a odds ratio function	Altınçay et al,[44] Liu et al,[46] and Ren et al[45]
TF. Gain ratio	TF-GR	Multiply TF by a gain ratio function	Quinlan et al[47]
TF. Correlation coefficient	TF-CC	Multiply TF by a correlation coefficient function	Liu et al[46] and Ren et al[45]
TF. Probability based	TF-PB	Multiply TF by a probability function	Liu et al[46]
TF. Relevance frequency	TF-RF	Multiply TF by the relevance frequency function	Lan et al[20]
TF. Inverse gravity moment	TF-IGM	Multiply TF by the inverse gravity moment function	Chen et al[2]

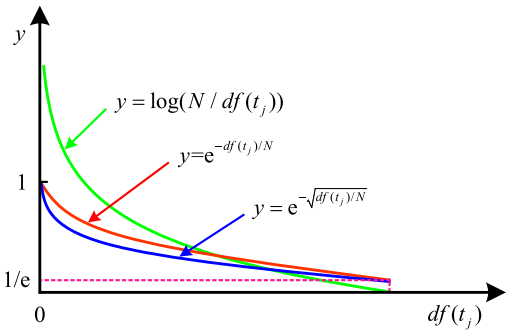

FIGURE 2 Qualitative comparison of IDF, IEF and RIEF

mainly considers three factors: TF, DF and normalization when calculating weights. Therefore, TF-IDF have incorporated with three different fundamental assumptions.[43] Firstly, TF assumption, which suggests that multiple occurrences of a term in a text are no less important than single occurrences. Secondly, IDF assumption, which means that rare terms are no less significant than frequent terms. Thirdly, normalization assumption, which indicates that long texts are no less important than short texts for the same quantity of term matching. Considering these assumptions, the improved weighting scheme we propose should also incorporate them.

In this work, being inspired from the previous studies (Table 1) and by considering such assumption, we naturally want to multiply the local TF-based component by the global log-like IDF component since the TF seems to fit naturally in text representation. In addition, further analysis of IDF in TF-IDF shows that the N can be regarded as a constant in a specific TC task, which denotes the total number of texts in the training set. Consequently, the $IDF(t_j)$ decreases as $df(t_j)$ increases, but the decrease is small, this means that the importance of a term relative to a document is inversely proportional to the frequency of appearance of this term in all the documents.[16] Their relation is illustrated in Figure 2, where the x-axis indicates the DF and the y-axis indicates the importance of a term. This behavior just reflects the assumption mentioned above, thus the IDF can be applied to make the local weighting factor (ie, TF) less insensitive.

Thus, in order to reduce the effect of term (feature) with high local weighting factor TF, meanwhile the importance of term (feature) with low DF can be further enhanced. In this study, we replaced the IDF factor (or logarithmic function) with a novel function (or new global weighting factor) that possesses the following two properties: (i) when the DF $df(t_j)$ of the term t_j changes, the value of the function has a larger attenuation rate, (ii) it should be a bounded function. It is well known, however, that the attenuation rate of exponential function is faster than the logarithmic function (shown in Figure 2), so the inverse function e^x of log$x$ is naturally chosen to construct the new function to fulfill the above goal. But, as shown in Figure 1, due to the original function log$x$ has the same monotonicity as its inverse function e^x, and the curves of this two functions are symmetrical with respect to $y = x$. Therefore, it is worth mentioning that this new function should also satisfy the following three requirements. First, the independent variable x must be the function of DF $df(t_j)$. Second, to avoid division by zero, the $df(t_j)$ cannot be used as a denominator. Third, the x value increases as $df(t_j)$ decreases, and vice versa.

According to these descriptions, we treat $x = -df(t_j)/N$ as an independent variable in which the $df(t_j)$ is utilized as numerator to avoid zero divisor. That is, the log probability ratio $\log(N/df(t_j))$ in TF-IDF (see (4)) is replaced by the exponential probability ratio $e^{-df(t_j)/N}$. In fact, $e^{-df(t_j)/N}$ is named inverse exponential function in mathematics. Hence, the improved term weighting scheme (denoted as $TFIEF(t_j, d_k)$) can be formulated as follows:

$$TFIEF(t_j, d_k) = \frac{tf(t_j, d_k) \cdot e^{-df(t_j)/N}}{\sqrt{\sum_{j=1}^{n} \left(tf(t_j, d_k) \cdot e^{-df(t_j)/N} \right)^2}} \tag{5}$$

In this work, the term weighting scheme expressed by (5) is named TF-IEF. Apparently, TF-IEF uses the novel global statistical model (ie, inverse exponential function) to characterize the global weighting factor like IDF in the corpus. Furthermore, $TFIEF(t_j, d_k)$ still embodies three different fundamental assumptions of $TFIDF(t_j, d_k)$ in which the $IEF(t_j)$ of the term t_j is considered as a new global weighting factor.

3.3 | Modified variations of TF-IEF scheme

It is clear that the proposed term weighting scheme TF-IEF is evolved from the TF-IDF, so it also consists of the local and global weighting factors, its correspond respectively to the TF component and IEF component. Therefore, we modify the initial TF-IEF term weights by replacing one of the components or both.

In fact, a term appears only once in a text is usually more than ten times as significant as a term appears ten times. Because of this, if the high local weighting factor (ie, TF) is reduced properly may obtain more reasonable term weighting, and then the performance of TC will be improved.[48] As described in Section 2.1, the TF factor has various variants by a logarithm operation. Here, an alternative approach is adopted in this study, that is, raw TF is square root (named RTF), namely replacing raw frequency $tf(t_j, d_k)$ with $\sqrt{tf(t_j, d_k)}$. In general, term weighting schemes using square root function–based TF factor is superior to the logarithmic function–based TF factor.[49] Besides, the inverse exponential frequency

TABLE 2 Quantitative comparison of IDF, IEF, and RIEF

	$df(t_j) = 0$	$df(t_j) = N$
$IDF(t_j)$	$\rightarrow +\infty$	=0
$IEF(t_j)$	=1	=1/e
$RIEF(t_j)$	=1	=1/e

($e^{-df(t_j)/N}$) in Equation (5) can be regarded as an adjustment coefficient to reduce TF appropriately. Meanwhile, note that the formula $df(t_j)/N$ is always less than 1 because the numerator $df(t_j)$ is always less than or equal to denominator N. For this reason, we compute the square root of IEF (named RIEF) to further reduce the value of $e^{-df(t_j)/N}$, namely replacing $e^{-df(t_j)/N}$ with $e^{-\sqrt{df(t_j)/N}}$. Eventually, the weight of term is significantly reduced, because the $tf(t_j, d_k)$ and $e^{-\sqrt{df(t_j)/N}}$ are multiplication operations where $e^{-\sqrt{df(t_j)/N}}$ is also always less than 1. It is necessary to emphasize that the attenuation rate of the $e^{-\sqrt{df(t_j)/N}}$ is faster than the $e^{-df(t_j)/N}$ (shown in Figure 2).

To sum up, three possible variations of TF-IEF formula can be represented as follows, respectively. In this research, these three term weighting schemes expressed by (6), (7), and (8) are, respectively, termed as RTF-IEF, TF-RIEF, and RTF-RIEF. Apparently, all these schemes are an improved version of TF-IEF.

$$RTFIEF\left(t_j, d_k\right) = \frac{\sqrt{tf\left(t_j, d_k\right)} \cdot e^{-df(t_j)/N}}{\sqrt{\sum_{j=1}^{n}\left(\sqrt{tf\left(t_j, d_k\right)} \cdot e^{-df(t_j)/N}\right)^2}}, \tag{6}$$

$$TFRIEF\left(t_j, d_k\right) = \frac{tf\left(t_j, d_k\right) \cdot e^{-\sqrt{df(t_j)/N}}}{\sqrt{\sum_{j=1}^{n}\left(tf\left(t_j, d_k\right) \cdot e^{-\sqrt{df(t_j)/N}}\right)^2}}, \tag{7}$$

$$RTFRIEF\left(t_j, d_k\right) = \frac{\sqrt{tf\left(t_j, d_k\right)} \cdot e^{-\sqrt{df(t_j)/N}}}{\sqrt{\sum_{j=1}^{n}\left(\sqrt{tf\left(t_j, d_k\right)} \cdot e^{-\sqrt{df(t_j)/N}}\right)^2}}. \tag{8}$$

3.4 | Qualitative and quantitative comparisons of IDF, IEF, and RIEF

Here, both qualitative and quantitative analyses are performed to compare the performance of IDF, IEF, and RIEF in term weighting. For qualitative analysis, these three global weighting factors are plotted in Figure 2. Through observing the initial attenuation rate of the three factors, RIEF factor achieves the fastest attenuation rate, IEF factor lags behind RIEF factor, and IDF factor is the worst. They are sorted as RIEF>IEF>IDF. Furthermore, only when the DF $df(t_j)$ is large enough to exceed a certain value, the attenuation rate of the IDF more faster than IEF and RIEF. More importantly, it is clear that the IDF function is unbounded, while the IEF and RIEF functions are bounded that better handles the coverage issue.

For quantitative analysis, let us consider two extreme cases, ie, DF $df(t_j) = 0$ and $df(t_j) = N$; the analysis results are presented in Table 2. As shown by Table 2, the global weighting factor of $IDF(t_j)$ falls in [0, +∞). In particular, if $df(t_j)$ is very small (even equal to zero if a certain term does not occur in a text), then the log probability ratio $\log(N/df(t_j))$ is close to +∞. In contrast, the global weighting factor $IEF(t_j)$ converges in an interval [1/e, 1]. Compared with the $IDF(t_j)$, the $IEF(t_j)$ does not appear two extremes (ie, +∞ and 0), which make the weight of term is quite meaningful whatever the value of $df(t_j)$ is. Actually, in the two extreme cases, the final weight $TFIEF(t_j, d_k)$ of term in the text can be considered to be determined by TF alone. Note that this observation is also consistent with the TF-RF by Lan et al.[20] Moreover, there is no doubt that RIEF (square root of IEF) yielded similar performances.

Through these analysis, we believe that the new global weighting factor IEF (or RIEF) not only has the capability to reduce significantly the high local weighting factor (ie, TF) in term weighting but also can overcome the limitations of the global weighting factors IDF in TF-IDF. Thus, it is more reasonable to use $TFIEF(t_j, d_k)$ or other improved versions to calculate the weight of terms in TC tasks, so as to generate a more representative feature (term). In Section 5, we present a detailed comparison of these schemes from different perspectives.

4 | EXPERIMENTAL SETUP

In this section, the experimental data sets are first briefly described and then introduces the fundamental approaches utilized in this study, including feature selection methods and classification algorithms. Finally, the evaluation of the performance measures used are also presented.

4.1 | Experimental data sets and preprocessing

An in-depth investigation is conducted to measure the performance of proposed term weighting methods (ie, TF-IEF and its various variants) with existing baseline approaches (ie, TF, TF-IDF, TF-CHI2, TF-IG, and TF-RF). Experiments are conducted on two distinct datasets (corpora), Reuters-21578 corpus and 20 Newsgroups corpus, which can be due to they have different characteristics. From Figure 3, it is noticeable that Reuters-21578 is a highly imbalanced (or skewed) data set, that is, numbers of texts in individual category are quite different.

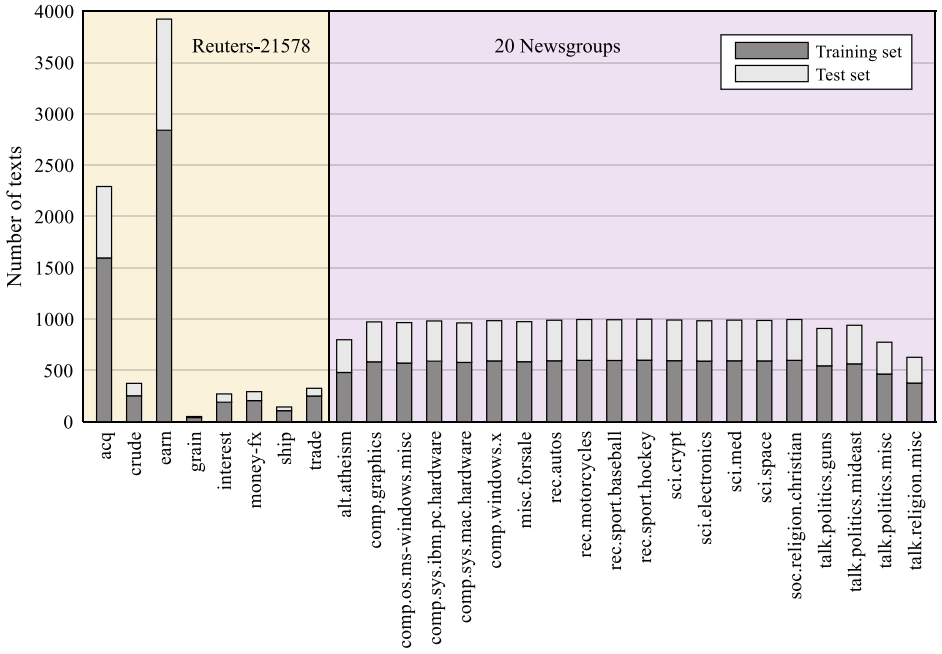

FIGURE 3 Text category statistics of datasets

In contrast, 20 Newsgroups is a slightly skewed data set, that is, numbers of texts almost evenly distributed among 20 different category. Most importantly, all these data sets can be downloaded directly from the Internet.*

4.1.1 | Reuters-21578 corpus

The first data set that we used in our experiment is the Reuters-21578 corpus, which is a popular dataset to use in the field of TC. 7674 texts from the top eight classes are selected in this study, which contains 5485 training set samples and 2189 test set samples.

The preprocessing phase may also affect the success of TC noticeably in addition to text representation.[6] Hence, punctuation marks, numbers, and other symbols are removed in the preprocessing step. Meanwhile, we discard terms that appear less than two times in the corpus to reduce the size of the feature set (or feature space), and consequently to decrease the computational cost. Furthermore, all letters are transformed to lowercase and removing the suffixes and prefixes from the terms using Porter's stemmer[50] in this study. For instance, ''works,'' ''working,'' and ''worked'' all have the same work root.

Finally, a total of 8786 distinct terms (features) are selected from the training set samples to build the initial feature set. Thereby, the document-term matrices of training set and test set are acquired at the end of preprocessing, which are 5485×8786 and 2189×8786, respectively.

4.1.2 | 20 newsgroups corpus

The second data set is another popular benchmark collection, namely, 20 Newsgroups corpus, which is consisted of 18 821 news articles distributed nearly evenly into 20 categories (shown in Figure 3). The 20 Newsgroups corpus have been partitioned into a training set of samples with 11 293 texts and a test set samples with 7528 texts. Similarly, the preprocessing method as described above is also employed to the 20 Newsgroups corpus. Finally, a total of 33 935 distinct terms (features) are selected from the training set samples to build the initial feature set. Thereby, the document-term matrices of training set and test set are acquired at the end of preprocessing, which are $11\,293 \times 33\,935$ and $7528 \times 33\,935$, respectively.

For simplicity, we termed the 20 classes alt.atheism, comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x, misc.forsale, rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, soc.religion.christian, talk.politics.guns, talk.politics.mideast, talk.politics.misc and talk.religion.misc as ''altath,'' ''compg,'' ''compomwm,'' ''compsiph,'' ''compsmh,'' ''compwx,'' ''miscf,'' ''recaut,'' ''recmot,'' ''recsb,'' ''recsh,'' ''scicry,'' ''sciele,'' ''scimed,'' ''scisspa,'' ''socrc,'' ''talkpg,'' ''talkpmid,'' ''talkpmis,'' and ''talkrmis,'' hereafter, respectively.

4.2 | Feature selection methods and their term weighting schemes

The main problem of TC is not only the excessive size of the data set,[51] but also the extremely high dimensional original feature space.[52,53] This is because most of the features are redundant or irrelevant to the TC task that may bring negative effects on the classifier.[53,54] Thus, it is essential

* http://ana.cachopo.org/datasets-for-single-label-text-categorization (Accessed on July 18, 2018).

TABLE 3 Fundamental information elements used for feature selection in TC

	In category c_i (c_i)	Not in category c_i ($\bar{c}_i$)
Containing feature t_j (t_j)	A_{ij}	C_{ij}
Not containing feature t_j ($\bar{t}_j$)	B_{ij}	D_{ij}

to reduce the original feature space without sacrificing the classification performance. Feature selection can help in building faster and accurate models for TC purposes.[54] Basically, feature selection techniques can be divided into three categories: the filter methods, the wrapper methods, and the embedded methods.[55] Among all these methods, filter-based approaches are generally separated into two distinct ways: locally and globally.[46,56] In local feature selection, each category is represented with a different set of features, whereas in global feature selection, the set of features is chosen under all categories.[51,56] In our study, the global feature selection method is used to select features.

Currently, the most widely used filter-based methods include the DF, Chi-square (CHI2), information gain (IG) and mutual information (MI) ones.[51,53,57] Among these techniques, CHI2 and IG have been proven to be the most effective methods in feature selection.[52] Thus, on the one hand, we adopt CHI2 as a feature selection approaches, on the other hand, we replaced the IDF function (see Equation (3) or (4)) with CHI2 and IG as term weighting schemes in this study, respectively. Furthermore, the newly proposed term weighting method based on relevance frequency is introduced. Mathematical backgrounds of these methods are presented in the following.

4.2.1 | Term weighting based on chi-square

CHI2 is one of the most popular feature selection methods for TC. The $\chi^2(t_j, c_i)$ statistic measures the lack of independence between a feature (term) t_j and a category c_i if it is assumed that the appearance of a term (feature) is actually independent of the category label.[53] The $\chi^2(t_j, c_i)$ value of feature t_j with respect to category c_i is defined in Equation (9) and its four fundamental information elements are listed in Table 3.[45,57]

$$\chi^2\left(t_j, c_i\right) = \frac{N \cdot \left(A_{ij} \cdot D_{ij} - B_{ij} \cdot C_{ij}\right)^2}{\left(A_{ij} + B_{ij}\right) \cdot \left(C_{ij} + D_{ij}\right) \cdot \left(A_{ij} + C_{ij}\right) \cdot \left(B_{ij} + D_{ij}\right)} \tag{9}$$

where N is the total number of documents in the training set, and it is the sum of A_{ij}, B_{ij}, C_{ij}, and D_{ij}. A_{ij} denotes the number of documents that contain feature t_j in category c_i, B_{ij} denotes the number of documents that do not contain feature t_j in category c_i, C_{ij} denotes the number of documents that contain feature t_j but do not belong to category c_i, and D_{ij} denotes the number of documents that do not contain feature t_j and do not belong to category c_i.

Then, by combining the TF (denoted as $tf(t_j, d_k)$) and Chi-square (denoted as $\chi^2(t_j, c_i)$), the TF-CHI2 weight of term t_j is defined by $TFCHI2(t_j, c_i) = tf(t_j, d_k) \cdot \chi^2(t_j, c_i)$.[43] Obviously, the TF-CHI2 of a term t_j in document d_k is the product of the TF and the Chi-square. Just like formula (4), to eliminate the effect of document length, the $TFCHI2(t_j, c_i)$ can be calculated with Equation (10) in this study.

$$TFCHI2\left(t_j, c_i\right) = \frac{tf\left(t_j, d_k\right) \cdot \chi^2\left(t_j, c_i\right)}{\sqrt{\sum_{j=1}^{n} \left(tf\left(t_j, d_k\right) \cdot \chi^2(t_j, c_i)\right)^2}} \tag{10}$$

4.2.2 | Term weighting based on information gain

IG measures the decrease in entropy when the presence or absence of a term (feature) contributes to make the correct classification decision on any category. IG reaches a maximum value if a certain term is an ideal indicator for assigning the document to any category.[58] With the above notation (as shown in Table 3), the IG of term t_j can be expressed as[20,59]

$$\begin{aligned} IG\left(t_j\right) = {} & \frac{A_{ij}}{N} \cdot \log \frac{A_{ij} \cdot N}{\left(A_{ij} + C_{ij}\right) \cdot \left(A_{ij} + B_{ij}\right)} + \frac{B_{ij}}{N} \cdot \log \frac{B_{ij} \cdot N}{\left(B_{ij} + D_{ij}\right) \cdot \left(A_{ij} + B_{ij}\right)} \\ & + \frac{C_{ij}}{N} \cdot \log \frac{C_{ij} \cdot N}{\left(A_{ij} + C_{ij}\right) \cdot \left(C_{ij} + D_{ij}\right)} + \frac{D_{ij}}{N} \cdot \log \frac{D_{ij} \cdot N}{\left(B_{ij} + D_{ij}\right) \cdot \left(C_{ij} + D_{ij}\right)}. \end{aligned} \tag{11}$$

Similar to the TF-CHI2 weight method, the TF-IG weight of term t_j can be defined by Lan et al[20] and Debole and Sebastiani.[43] In this equation, we used the normalization formula (12) in order to eliminate the effect of document length.

$$TFIG\left(t_j\right) = \frac{tf\left(t_j, d_k\right) \cdot IG\left(t_j\right)}{\sqrt{\sum_{j=1}^{n} \left(tf\left(t_j, d_k\right) \cdot IG\left(t_j\right)\right)^2}} \tag{12}$$

4.2.3 | Term weighting based on relevance frequency

Unlike the above supervised term weighting methods (ie, TF-CHI2 and TF-IG), Lan et al[20] proposed a novel supervised term weighting scheme, TF-RF, which takes into account only the frequency of relevant documents. The TF-RF weight of term t_j with respect to category c_i can be calculated as shown in Equation (13) using RF.[2,20,38]

$$TFRF\left(t_j, c_i\right) = tf\left(t_j, d_k\right) \cdot \log_2\left(2 + \frac{A_{ij}}{\max\left(1, C_{ij}\right)}\right) \tag{13}$$

We can see from Equation (13) that the IDF part of TF-IDF (shown in Equation (3)) is actually replaced by RF (denoted as $RF(t_j, c_i) = \log_2(2+A_{ij}/\max(1, C_{ij}))$). In this formula, the base of this logarithmic operation is 2, so the constant value 2 is added to the formula to avoid zero weight. To avoid division by 0, set the minimal denominator as 1 in Equation (13). Similarly, we use the following normalized $TFRF(t_j, c_i)$ measure in this study.

$$TFRF\left(t_j, c_i\right) = \frac{tf\left(t_j, d_k\right) \cdot RF\left(t_j, c_i\right)}{\sqrt{\sum_{j=1}^{n}\left(tf\left(t_j, d_k\right) \cdot RF(t_j, c_i)\right)^2}} \tag{14}$$

4.3 | Classification algorithms

To prove the classification performance of the proposed term weighting methods over existing weighting schemes, it is necessary to apply the classifiers widely used for TC tasks. For this purpose, some classifiers such as Naïve Bayes (NB) and SVM are utilized in this research. A brief description about these two classifiers is given in the following.

4.3.1 | Naïve Bayes

The NB classifier is an oldest formal classification algorithms based on Bayes' theorem, which regards the features as independent from each other.[45] Since NB classifier computational cost is very low, its have been widely used for TC tasks.

For TC, we assume that the document d_k consisting of a number of term is written as $d_k = (t_1, t_2, \ldots, t_n)$. The probability that a document d_k belongs to the category c_i can be defined by Equation (15).[9,21] For more details, you can refer to Farid et al,[60] which gives a complete description of the theory of NB.

$$P(c_i|d_k) = \frac{P(c_i)\prod_{j=1}^{n} P\left(t_{jk}|c_i\right)}{P(d_k)} \tag{15}$$

4.3.2 | Support vector machine

The SVM is first proposed by Vapnik,[61] which can generally be divided into two categories, such as linear and nonlinear, according to the different kernel functions. In this research, linear SVM is employed because it has been proved to be perform better than the nonlinear.[20] As a text classifier, it has many advantages, such as the redundant features and high dimensional features are well handled by using the kernel function.[53]

Assume that a binary classification task: $\{(x_i, y_i)\}$, $x_i \in R^m$, $y_i \in \{-1, +1\}$ and $i = 1, \ldots, m$, in which x_i are data points and y_i are the corresponding labels output for the ith training sample. Then, the output of a linear SVM is defined as $w^{\mathrm{T}} \cdot x+b = 0$, where w is an n-dimensional coefficient vector and b is the offset from the origin that is determined by the training process. There are many hyperplanes that can separate the positive examples from negative examples in the n-dimensional space with the maximum margin, which can be expressed as an optimization problem formula (16), by solving it to find the optimal hyperplane with the maximum margin. More details about the SVM can be referred to the work of Vapnik.[61]

$$\begin{aligned} &\min_{w,b} && \frac{1}{2}w^{\mathrm{T}}w \\ &\text{subject to}: && y_i\left(w \cdot x_i + b\right) \geq 1 \end{aligned} \tag{16}$$

Furthermore, it is worth mentioning that we use grid-search strategy and ten-fold cross validation method to find out the optimal parameters α and C for both NB and SVM classifiers in this research, where α is the smoothing parameter of NB and C is the penalty parameter of SVM. Other parameters are left at default values. The classifier parameters considered in the benchmarking experiments are summarized in Table 4.

4.4 | Evaluation of the performance

The macro and micro values of F1 measure are usually employed to evaluate the classification performance of TC results.[53,62] To fully define these two measures, three basic measures should be first defined, namely precision, recall, and the F1 measure. This measures are define according to a contingency table of predictions for a target category c_i, and its components are shown in Table 5.[21] The precision ($P(c_i)$) is the proportion of

TABLE 4 Classifier parameters

Weighting scheme	Classifier	Parameters	
		Reuters-21578	20 Newsgroups
TF	NB	α=0.01	α=0.01
	SVM	$C = 2$	$C = 5$
TF-IDF	NB	α=0.1	α=0.01
	SVM	$C = 1$	$C = 2$
TF-CHI2	NB	α=0.001	α=0.0001
	SVM	$C = 5$	$C = 7$
TF-IG	NB	α=0.0001	α=0.0001
	SVM	$C = 1$	$C = 7$
TF-RF	NB	α=0.01	α=0.01
	SVM	$C = 2$	$C = 3$
TF-IEF	NB	α=0.01	α=0.01
	SVM	$C = 2$	$C = 4$
RTF-IEF	NB	α=0.01	α=0.01
	SVM	$C = 1$	$C = 3$
TF-RIEF	NB	α=0.1	α=0.01
	SVM	$C = 2$	$C = 2$
RTF-RIEF	NB	α=0.01	α=0.01
	SVM	$C = 1$	$C = 3$

TABLE 5 Contingency table for category c_i

	True label c_i	True not c_i
Predicted label c_i	True positive (TP)	False positive (FP)
Predicted not c_i	False negative (FN)	True negative (TN)

correct assignments among all the test documents that should be assigned to the target category c_i. The recall ($R(c_i)$) is the proportion of correct assignments among all the test documents assigned to the target category c_i. The F1 measure ($F_1(c_i)$) is the harmonic mean of the $P(c_i)$ and $R(c_i)$. Thus, the precision ($P(c_i)$), recall ($R(c_i)$), and the F1 measure ($F_1(c_i)$) are defined as follows:

$$P(c_i) = \frac{TP(c_i)}{TP(c_i) + FP(c_i)}, \tag{17}$$

$$R(c_i) = \frac{TP(c_i)}{TP(c_i) + FN(c_i)}, \tag{18}$$

$$F_1(c_i) = \frac{2 \cdot P(c_i) \cdot R(c_i)}{P(c_i) + R(c_i)} = \frac{2 \cdot TP(c_i)}{2 \cdot TP(c_i) + FP(c_i) + FN(c_i)}. \tag{19}$$

In the contingency table, the true positive (TP) is the count of test documents that belong to category c_i correctly classified into category c_i, false positive (FP) is the count of test documents that do not belong to category c_i incorrectly classified into category c_i, false negative (FN) is the count of test documents that belong to category c_i but are not classified into category c_i, and true negative (TN) is the count of test documents do not belong to category c_i and meanwhile are not classified into category c_i.

After that, the macro-F1 and micro-F1[21] can be formulated as Equations (20) and (21), respectively. It is noticeable that the macro-F1 is calculated for each category and then averaged over all categories. While the micro-F1 is calculated globally over all categories without class discrimination.

$$\text{macro-}F1 = \frac{1}{m}\sum_{i=1}^{m} F_1(c_i), \tag{20}$$

$$\text{micro-}F1 = \frac{2 \cdot \sum_{i=1}^{m} TP(c_i)}{2 \cdot \sum_{i=1}^{m} TP(c_i) + \sum_{i=1}^{m} FP(c_i) + \sum_{i=1}^{m} FN(c_i)}, \tag{21}$$

where m is the total number of categories.

5 | EXPERIMENTS AND ANALYSIS OF RESULTS

To validate the classification performance of proposed term weighting schemes with existing methods, we carried out extensive experiments of TC using nine term weighting methods, which are mentioned in previous section. All nine weighting methods to be compared in the experiments and their mathematic formations are shown in Table 6.

TABLE 6 All nine weighting methods to be compared in the experiments

ID	Weighting method	Mathematical formations
1	TF	Equation (1)
2	TF-IDF	Equation (4)
3	TF-CHI2	Equation (10)
4	TF-IG	Equation (12)
5	TF-RF	Equation (14)
6	TF-IEF	Equation (5)
7	RTF-IEF	Equation (6)
8	TF-RIEF	Equation (7)
9	RTF-RIEF	Equation (8)

5.1 | Results on Reuters-21578 corpus

Figures 4 and 5 report the experimental results of TC on the Reuters-21578 corpus using the NB and SVM classifiers, respectively. It can be seen from Figure 4 that the performance of different schemes at a small feature sets (ie, less than 250 features) are relatively low especially for macro-F1 measure, the reason is that 250 or fewer features are not enough to allow an effective discrimination amongst the many texts in different classes.[57] Therefore, the increment in performance is noticeable when we increased the feature sets, since a larger number of features are more likely to cover all the texts' classes,[57] it makes different classes of texts can be discriminated better. In Figure 4, we see the TF-IDF scheme is the one that offers the lowest performance for smaller feature sets (ie, less than 400 features), but it improves substantially as the number of feature sets increases. In particular, when the number of feature sets falls in [750, 2000], TF-IDF is consistently superior to the TF-CHI2, TF-IG and TF-RF schemes. Even so, the term weighting schemes (ie, TF-IEF, RTF-IEF, TF-RIEF, and RTF-RIEF) we proposed are consistently higher performance over the TF-IDF scheme in almost the entire range of feature sets.

Furthermore, in terms of macro-F1, the performances of the TF-RF scheme consistently outperforms the TF-CHI2 and TF-IG when the number of features exceeds 250. Similarly, in terms of micro-F1, the performance of the TF-RF scheme consistently outperforms the TF-CHI2 and TF-IG for all feature sizes. Besides, it is necessary to note that TF-IEF performs comparatively well than all other schemes in most cases with more than 500 features.

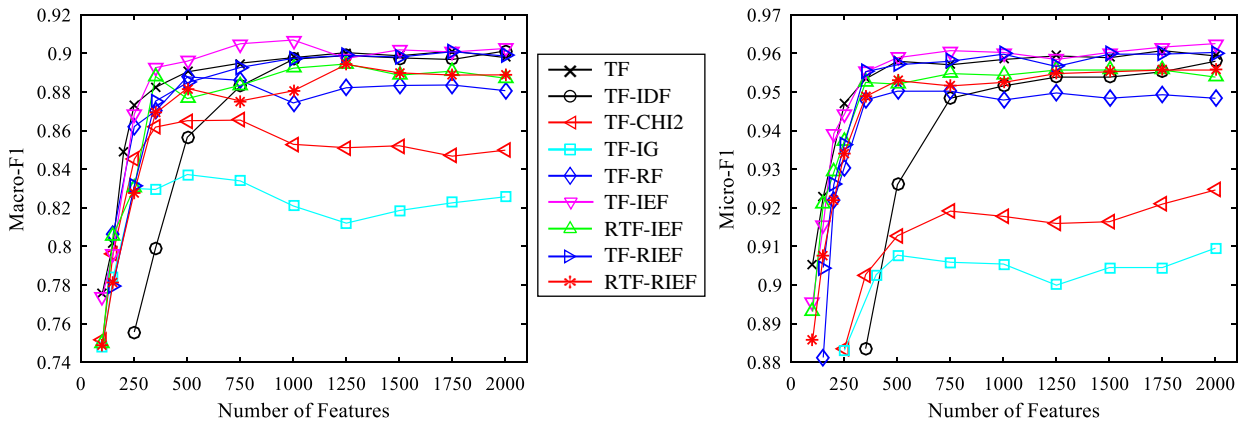

FIGURE 4 Macro-F1 and micro-F1 measure with different numbers of features on the Reuters-21578 corpus using NB classifier

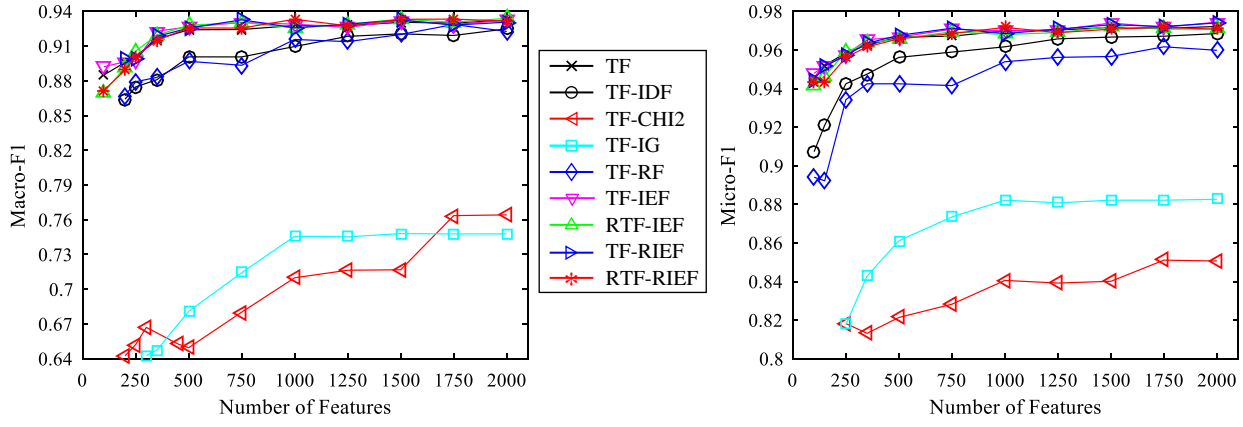

FIGURE 5 Macro-F1 and micro-F1 measure with different numbers of features on the Reuters-21578 corpus using SVM classifier

Although the distribution of texts in the Reuters-21578 data set is highly imbalanced (see Figure 3), when using SVM classifier to classify text in Reuters-21578 data set, the performance in Figure 5 are higher than those in Figure 4 for different term weighting schemes. In terms of macro-F1 and micro-F1, the performance of TF-CHI2 and TF-IG are far worse than other term weighting schemes in the complete range of different number of features. But in fact, the macro-F1 and micro-F1 results produced by TF-IG method generally outperforms TF-CHI2 method on the Reuters-21578 corpus with a linear SVM classifier. Furthermore, it can be said that the macro-F1 scores of TF-IDF and TF-RF are very close to each other. It must be noted, however, the TF-IDF scheme shows slightly better performance than TF-RF scheme in terms of micro-F1 results on Reuters-21578 with a linear SVM classifier. Apart from these, for any number of features, four of the proposed weighting schemes TF-IEF, RTF-IEF, TF-RIEF, and RTF-RIEF show obvious advantages over the TF-IDF,TF-CHI2, TF-IG and TF-RF in terms of both macro-F1 and micro-F1.

5.2 | Results on 20 newsgroups corpus

Figures 6 and 7 show the experimental results of TC on the 20 Newsgroups corpus using the NB and SVM classifiers, respectively/ Similarly to the results of Reuters-21578 corpus (shown in Figure 4 and Figure 5), from Figure 6 it is seen in general that, the macro-F1 and micro-F1 performance of all weighting schemes on the 20 Newsgroups corpus are increasing as the number of features increases.

From Figure 6, we can easily find that the TF-CHI2 and TF-IG schemes are also obviously inferior to all other term weighting schemes over the whole range of different number of features. However, in fact, in terms of both macro-F1 and micro-F1, the TF-CHI2 is slightly better than TF-IG scheme. Moreover, for the rest seven term weighting schemes such as TF, TF-IDF, TF-RF, TF-IEF, RTF-IEF, TF-RIEF, and RTF-RIEF, their macro-F1 and micro-F1 measures are not much different. Even so, the macro-F1 and micro-F1 measures produced by proposed weighting schemes, especially RTF-RIEF, TF-RIEF, and RTF-IEF are outperforming other term weighting schemes a bit in most cases.

However, for TC using SVM classifier on the 20 Newsgroups data set, the classification performance differences among the nine methods shown in Figure 7 are greater than those in the NB case shown in Figure 6. Obviously, TF-CHI2 and TF-IG scheme still achieves the worst performance among the nine term weighting schemes over the full range of different number of features. More precisely, TF-CHI2 scheme shows slightly better performance than the TF-IG scheme. In addition, it is clear that the performances of TF-RF weighting schemes for the SVM classifier is lower than other six schemes. From Figure 7, we can also see that, in terms of macro-F1, TF-RIEF outperforms TF-IDF a bit when the number of features is less than 1200. A similar phenomenon can also be seen in terms of micro-F1, ie, TF-RIEF outperforms TF-IDF a bit when the number of features is less than 2000. Moreover, the macro-F1 and micro-F1 results produced by TF, TF-IEF, RTF-IEF, and RTF-RIEF

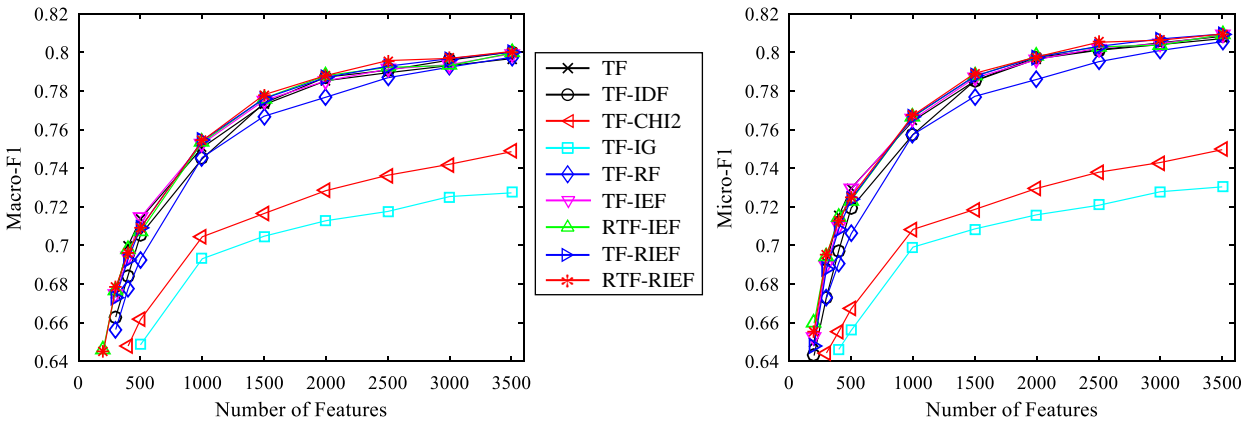

FIGURE 6 Macro-F1 and micro-F1 measure with different numbers of features on the 20 Newsgroups corpus using NB classifier

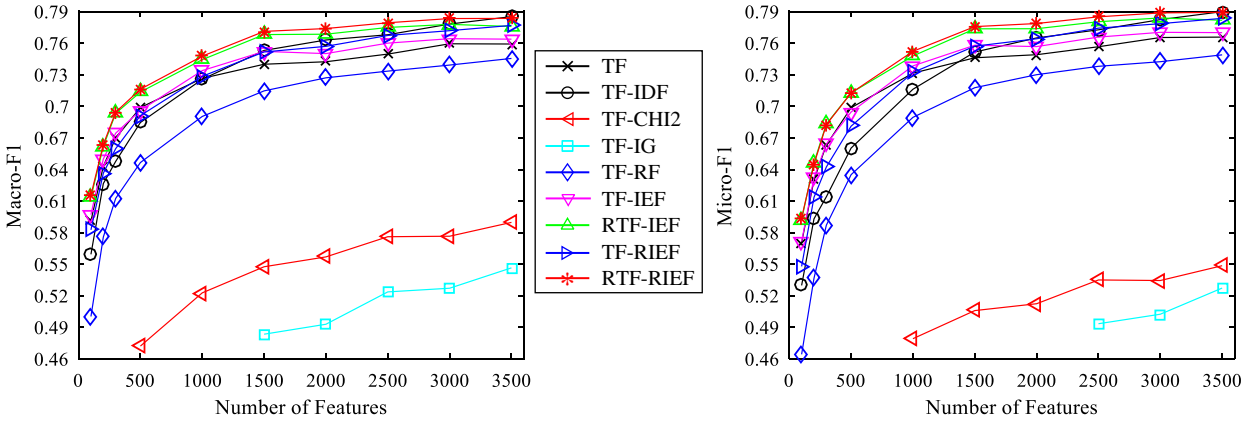

FIGURE 7 Macro-F1 and micro-F1 measure with different numbers of features on the 20 Newsgroups corpus using SVM classifier

can be ranked as RTF-RIEF>RTF-IEF > TF-IEF > TF for a large number of features (ie, greater than 800). Actually, in almost the whole range of features, the classification performance reported in Figure 7 clearly reveals that the proposed RTF-RIEF and RTF-IEF term weighting schemes are obviously superior to all other schemes in terms of both macro-F1 and micro-F1. Besides that, the RTF-RIEF outperforms RTF-IEF a bit.

5.3 | Statistical significance tests

In spite of the experimental results depicted in Figures 4 to 7 clearly show that at least one of the proposed term weighting schemes outperform other schemes. The significance tests is needed to be conducted to illustrate more clearly the differences performance of our presented TF-IEF, and its various variants schemes perform better than other schemes. Consequently, the analysis of variance (ANOVA) and paired-sample *t*-test are used in micro-F1 performance based on the SVM classifier, since it provides the best classification performance in TC tasks.

For this purpose, the TC is implemented according to the feature sets of different sizes for each weighting scheme. Thus, on the one hand, 15 feature sets are built for Reuters-21578 data set where the number of features started with 100 and ended with 2000. In addition, for the first nine feature sets, the number of features is increased by 50, and then for the last six feature sets, the number of features is increased by 250. On the other hand, 11 feature sets are created for 20 Newsgroups dataset where the number of features started with 100 and ended with 3500. Furthermore, for the first first feature sets, the number of features is increased by 100, and then for the last six feature sets, the number of features is increased by 500. The statistical test results are listed in Table 7.

In Table 7, F-crit represents the critical value of *F*. ANOVA reveals that this differences in micro-F1 performance are statistically significant for all corpora. This is evident from the F-value is always greater than F-crit and a low probability value ($P < 0.01$) over both data sets. The maximum averages among the micro-F1 scores achieved by different term weighting schemes are highlighted in bold, and the secondary maximum averages are underlined in Table 7. Considering the maximum averages, our proposed schemes TF-IEF, RTF-IEF, TF-RIEF, and RTF-RIEF significantly outperform other weighting schemes in average.

To visually illustrate the differences performance of the nine term weighting schemes, their average performances and standard deviations are plotted in Figure 8 through the results of the ANOVA. We can see from Figure 8 that the average performances given by the nine term weighting schemes are quite different. More importantly, the average performances is totally inconsistent with the standard deviations for different term

TABLE 7 Results of ANOVA

Weighting scheme	Reuters-21578		20 Newsgroups	
	Average	Standard deviation	Average	Standard deviation
TF	0.963667	0.007797	0.705161	0.063632
TF-IDF	0.949536	0.017972	0.691902	0.089487
TF-CHI2	0.818517	0.027184	0.452106	0.088352
TF-IG	0.845653	0.042000	0.405504	0.090482
TF-RF	0.938267	0.022045	0.654876	0.095488
TF-IEF	**0.964702**	0.008124	0.709086	0.066483
RTF-IEF	0.962388	0.009418	<u>0.724990</u>	0.064288
TF-RIEF	<u>0.964215</u>	0.008678	0.703616	0.079044
RTF-RIEF	0.962144	0.010050	**0.727406**	0.066506
F-value	116.5161	—	26.57792	—
P-value	1.89E-54[a]	—	1.23E-20[a]	—
F-crit	2.655491	—	2.715364	—

[a]Significant at 0.01 level.

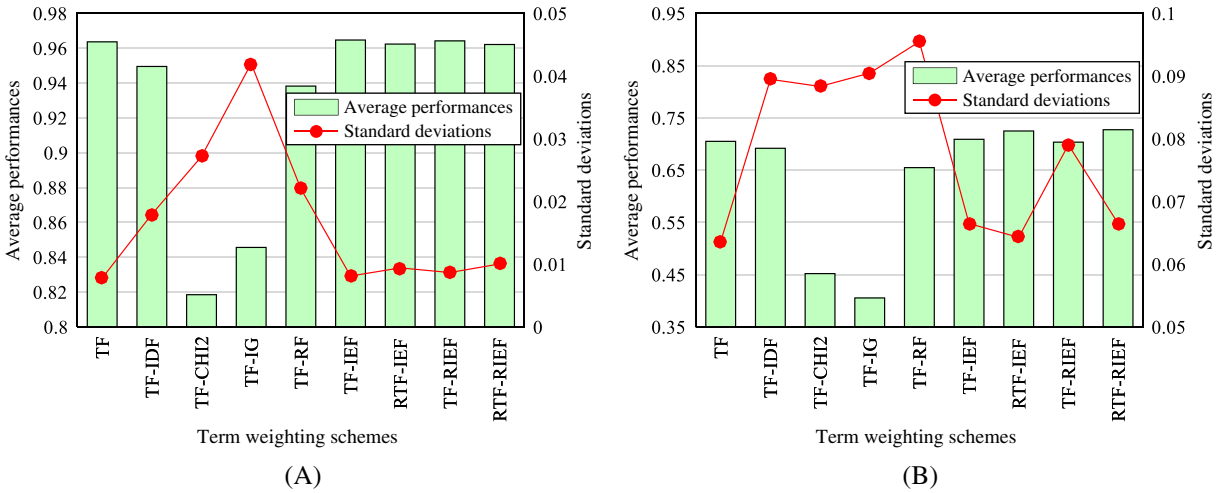

FIGURE 8 Comparison of average performances and standard deviations for all corpora. A, Reuters-21578; B, 20 Newsgroups

weighting methods as shown in Figure 8. It is worth to mention that the standard deviations with indicates the stableness of the schemes, which means the proposed term weighting schemes such as TF-IEF, RTF-IEF, TF-RIEF, and RTF-RIEF can be achieved the better average performances without sacrificing the stable performance for all corpora. The reasoning is that this four schemes adopts IEF or RIEF to substitute IDF factor that reduce significantly the high TF factor in term weighting.

It is necessary to emphasize that the performance (ie, macro-F1 and micro-F1 values) of all term weighting schemes including the proposed ones based on Reuters-21578 data set is higher than the performance based on 20 Newsgroups data set. This is due in part to the fact that different characteristic (see Figure 3) of document may have effects on term weighting and classification performance. For instance, the highly skewed in the Reuters-21578 data set, and moreover, the large number of categories in the 20 Newsgroups data set and some text categories are very similar. More detailed classification results regarding those behaviors are provided in Figures 9 and 10. In the graphs, the vertical axis of the confusion matrix indicates the actual label of classification, and the horizontal axis corresponds to the predicted label of classification.

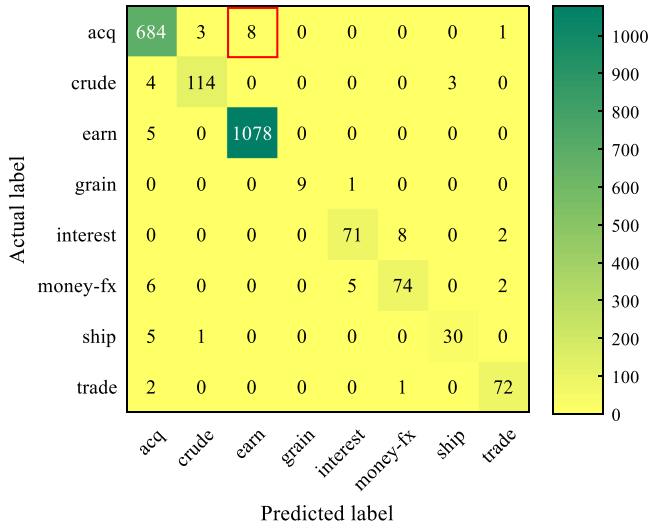

FIGURE 9 Confusion matrix of the proposed scheme TF-IEF on the Reuters-21578 corpus using SVM

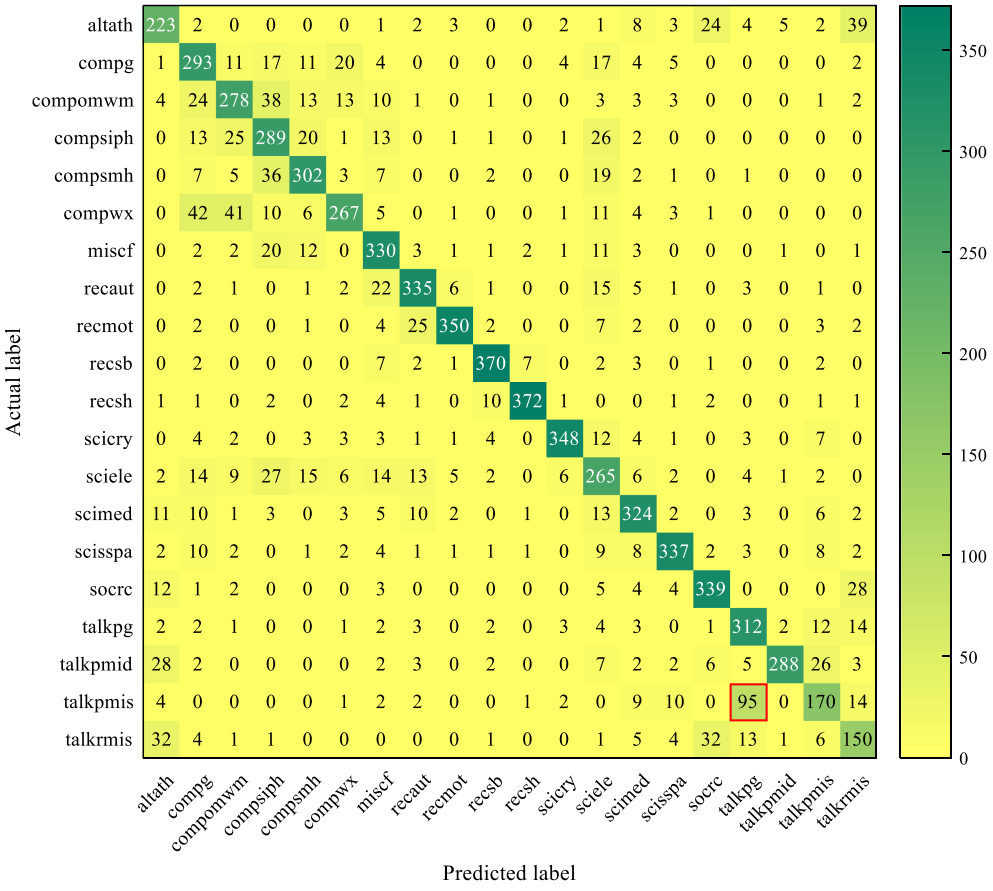

FIGURE 10 Confusion matrix of the proposed scheme RTF-RIEF on the 20 Newsgroups corpus using SVM

Furthermore, the value at the diagonal line represents the correct number of classification for each class of the test set, whereas the value at the nondiagonal line represents the incorrect number of classification for each class of the test set.

As pointed out in the previous observation, since the performances of TF-IEF and RTF-RIEF are better than other term weighting methods for the Reuters-21578 and 20 Newsgroups corpus respectively, it is reasonable that the TF-IEF and RTF-RIEF are selected for the TC experiments. We can see from Figure 9 that SVM performs better on the representation of proposed scheme TF-IEF in spite of being highly imbalanced in the Reuters-21578 data set. This suggests that there are many values equal to zeros outside the diagonal line. Meanwhile, it is clear to see that the number of positive classes are grouped into the negative class is very small. For example, if the actual label is "acq", the number of documents misclassified into "earn" is equal to 8 (shown in Figure 9), which is the highest value in the Reuters-21578 data set.

In contrast, the documents in 20 Newsgroups are almost uniformly distributed, but there are many positive classes that are incorrectly divided into negative classes. This means that there are many values that are not equal to zeros outside the diagonal line (shown in Figure 10). Besides, since the "compsmh" and "compsiph" classes, the "talkpmis" and "talkpg" classes, and so on are noticed to be quite similar in which many terms may overlap with each other. For this reason, it is more difficult to distinguish the features (terms) between different categories. For instance, when the actual label is "talkpmis", the number of documents misclassified into "talkpg" is equal to 95 (shown in Figure 10), which is the highest value in the 20 Newsgroups data set.

Nevertheless, further analysis on these significant results will be carried out to find those pairs of term weighting schemes whose performance differs significantly. In this study, the paired-sample *t*-test is employed to compare the difference of any two schemes out of nine term weighting schemes. It should be noted that the paired-sample *t*-test is a symmetric test, so the order of this test is not important[21] (eg, TF versus TF-IDF is the same as TF-IDF versus TF). Finally, the number of pairs that can be obtained from nine term weighting schemes is 36 pairs in our experiments. Table 8 presents the results achieved, where the T-crit represents the critical value of *T*.

TABLE 8 Results of paired-sample *t*-test

Pair		Reuters-21578			20 Newsgroups		
		T-value	P-value	T-crit	T-value	P-value	T-crit
TF vs.	TF-IDF	2.792005	0.005812[a]	2.539483	0.400502	0.346750	2.552380
	TF-CHI2	19.87263	5.28E-13[a]	2.583487	7.708434	2.07E-07[a]	2.55238
	TF-IG	10.69892	1.02E-08[a]	2.60248	8.984771	2.26E-08[a]	2.55238
	TF-RF	4.206775	0.000296[a]	2.566934	1.453435	0.082157	2.566934
	TF-IEF	-0.35605	0.362235	2.467140	-0.14144	0.444467	2.527977
	RTF-IEF	0.405130	0.344287	2.472660	-0.72706	0.237808	2.527977
	TF-RIEF	-0.18198	0.428454	2.467140	0.050521	0.480117	2.539483
	RTF-RIEF	0.462983	0.323614	2.478630	-0.80154	0.216121	2.527977
TF-IDF vs.	TF-CHI2	15.56518	2.42E-14[a]	2.492159	6.324395	1.79E-06[a]	2.527977
	TF-IG	8.805671	1.96E-08[a]	2.539483	7.464106	1.67E-07[a]	2.527977
	TF-RF	1.534018	0.068331	2.47266	0.938361	0.179627	2.527977
	TF-IEF	-2.97661	**0.003876**[a]	2.539483	-0.51124	0.307699	2.552380
	RTF-IEF	-2.45201	0.011520	2.517648	-0.99597	0.166233	2.552380
	TF-RIEF	-2.84737	**0.004978**[a]	2.527977	-0.32538	0.374136	2.527977
	RTF-RIEF	-2.36919	0.013512	2.508325	-1.05613	0.152439	2.552380
TF-CHI2 vs.	TF-IG	-2.1004	0.023191	2.492159	1.222199	0.117921	2.527977
	TF-RF	-13.2492	1.25E-13[a]	2.47266	-5.16958	2.33E-05[a]	2.527977
	TF-IEF	-19.95	**4.97E-13**[a]	2.583487	-7.70824	**1.45E-07**[a]	2.539483
	RTF-IEF	-19.3636	**2.54E-13**[a]	2.566934	-8.28317	**7.44E-08**[a]	2.55238
	TF-RIEF	-19.7703	**1.81E-13**[a]	2.566934	-7.03643	**3.99E-07**[a]	2.527977
	RTF-RIEF	-19.1838	**9.88E-14**[a]	2.55238	-8.25682	**5.22E-08**[a]	2.539483
TF-IG vs.	TF-RF	-7.56159	1.00E-07[a]	2.517648	-6.28732	1.94E-06[a]	2.527977
	TF-IEF	-10.7775	**9.25E-09**[a]	2.60248	-8.96744	**2.32E-08**[a]	2.55238
	RTF-IEF	-10.5032	**1.3E-08**[a]	2.60248	-9.54661	**9.08E-09**[a]	2.55238
	TF-RIEF	-10.7064	**1.01E-08**[a]	2.60248	-8.22941	**3.76E-08**[a]	2.527977
	RTF-RIEF	-10.4454	**7.47E-09**[a]	2.583487	-9.50755	**9.66E-09**[a]	2.55238
TF-RF vs.	**TF-IEF**	-4.3577	**0.00019**[a]	2.55238	-1.54525	0.069843	2.55238
	RTF-IEF	-3.89695	**0.000485**[a]	2.539483	-2.02014	0.029253	2.55238
	TF-RIEF	-4.24199	**0.000245**[a]	2.55238	-1.30406	0.103896	2.539483
	RTF-RIEF	-3.81554	**0.000541**[a]	2.527977	-2.06723	0.026701	2.55238
TF-IEF vs.	RTF-IEF	0.720725	0.238637	2.472660	-0.57036	0.287394	2.527977
	TF-RIEF	0.158772	0.437494	2.467140	0.175660	0.431210	2.539483
	RTF-RIEF	0.765696	0.225250	2.472660	-0.64611	0.262776	2.527977
RTF-IEF vs.	TF-RIEF	-0.55271	0.292423	2.467140	0.695785	0.247493	2.539483
	RTF-RIEF	0.068439	0.472961	2.467140	-0.08660	0.465925	2.527977
TF-RIEF vs.	RTF-RIEF	0.603409	0.275636	2.472660	-0.76381	0.227180	2.539483

[a]Significant at 0.01 level.

It is apparent from this table that our proposed schemes TF-IEF, RTF-IEF, TF-RIEF, and RTF-RIEF seem more outstanding than (as shown in Table 7) other term weighting schemes for different data sets. The main reason is that the *P*-values of the *t*-test are in general very small (the bold rows in Table 8). However, it is also necessary to note that the difference in the performance of introduced term weighting schemes is not significant in all pairs (the underlined rows in Table 8). That is, the difference of performance between TF-IEF on one hand and RTF-IEF, TF-RIEF, and RTF-RIEF on the other hand, is not significant at 0.01 level, since the *P*-values are always greater than 0.01. But in fact, it is significant in comparison to other term weighting schemes especially TF-CHI2 and TF-IG.

5.4 | Similarity analysis of term weighting

For the purpose of the similarity analysis of term (feature) weighting, the text representation is carried out based on the features of same sizes for each data set and weighting scheme. First, top-50 terms (features) are selected by CHI2. Second, in order to ensure diversity in the data sets, we generate them from different classes. Hence, 10% of the text for each class are randomly selected for each dataset in this experiments, that is to say, 549 and 1129 texts are constructed for the Reuters-21578 and 20 Newsgroups corpus, respectively. Finally, the texts are then represented by the nine term weighting schemes. Results of this analysis are reported in Figures 11 and 12. In the graphs, each color represents a different category. Overall, we can easily identify the high density of color dots at the bottom of term (feature) weighting regardless of the category.

In Figure 11, small length vertical lines can be noticed when the sequence index of terms (features) is around 10, 28, and 40, which suggests that the same features (terms) can be extracted though they are represented by different weighting schemes. This similar behavior of Reuters-21578 is absent in the 20 Newsgroups corpus. On the contrary, a larger scatter in the 20 Newsgroups corpus of color dots can be observed compared

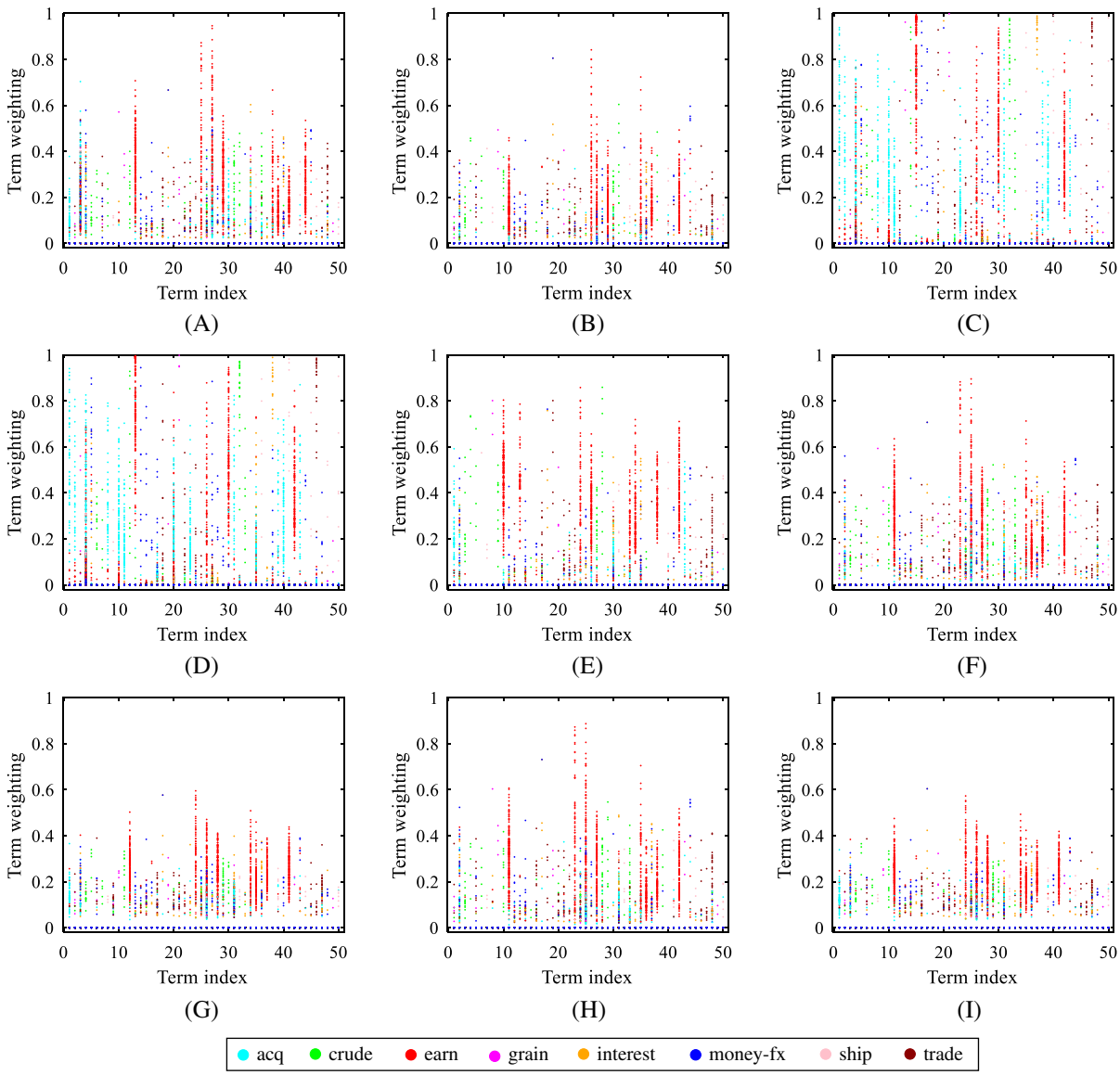

FIGURE 11 The term (feature) weighting distributions of top-50 terms of Reuters-21578 corpus. A,TF; B,TF-IDF; C,TF-CHI2; D,TF-IG; E,TF-RF; F,TF-IEF; G,RTF-IEF; H,TF-RIEF; I,RTF-RIEF

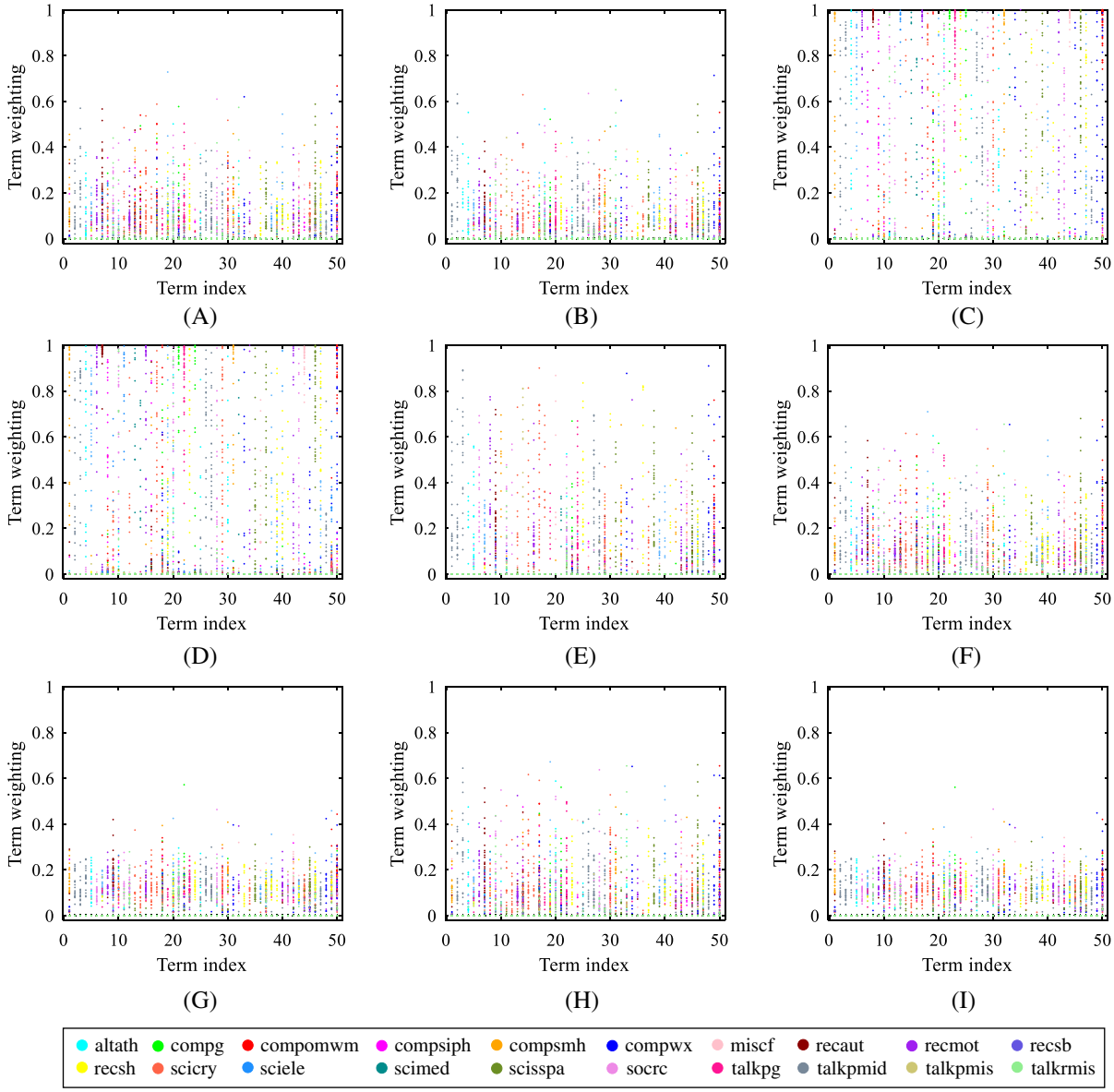

FIGURE 12 The term (feature) weighting distributions of top-50 terms of 20 Newsgroups corpus. A,TF; B,TF-IDF; C,TF-CHI2; D,TF-IG; E,TF-RF; F,TF-IEF; G,RTF-IEF; H,TF-RIEF; I,RTF-RIEF

to Reuters-21578 corpus. This means that the term (feature) weighting has relatively lesser concentration of color dots at the top in addition to TF-CHI2 and TF-IG (shown in Figure 12). However, it is necessary to emphasize that mostly there are term weighting by RTF-IEF and RTF-RIEF falls in [0, 0.3] regardless of the corpus, which means that they are comparatively better in text representation across wider range of classes. In other words, RTF-IEF and RTF-RIEF may generate more informative terms than other schemes in their top-*N* features (at least in the top-50 features).

Further analysis on the term (feature) weighting distributions revealed that this advantages can be partly attributed for their square root of TF (named RTF), which results in the term (feature) weighting equivalent to or sometimes even less than the TF. The main reason is that in spite of the TF is a common component in the nine term weighting schemes, we can note that the term (feature) weighting distributions of TF-IEF and RTF-IEF are quite different. Similarly, compared with the TF-RIEF and RTF-RIEF, we can note that their term (feature) weighting distributions are also quite different. So, this phenomenon can support the above conclusion. It is more important that the above similarity analysis of term (feature) weighting highlights the ability of our proposed schemes; in particular, RTF-IEF and RTF-RIEF can achieve the best scatter of terms (features) among the categories.

As a result, the analysis of above all experimental results demonstrated that the proposed term weighting schemes (ie, TF-IEF, RTF-IEF, TF-RIEF, and RTF-RIEF) can be caused an improvement on the TC performance regardless of the corpus. In particular, in almost all the cases, the RTF-RIEF is the best scheme among the proposed schemes. Hence, it can be concluded that the proposed global weighting factor IEF (or RIEF) is a good discriminator.

6 | CONCLUSIONS

In this work, under the three different fundamental assumptions (ie, TF assumption, IDF assumption, and normalization assumption) of TF-IDF, an improved TF-IEF weighting scheme and three modified schemes for TC are introduced. To offset the two deficiencies of TF-IDF, TF-IEF adopts a new global statistical method called IEF to substitute the global weighting factor IDF. As a result, the high local weighting factor TF is significantly reduced in term weighting, and then more informative terms are generated. To evaluate the classification performances of our methods such as TF-IEF, RTF-IEF, TF-RIEF, and RTF-RIEF, two published data sets (ie, Reuters-21578 corpus and 20 Newsgroups corpus) were utilized, and the performances of proposed schemes were compared to the existing schemes, namely TF, TF-IDF, TF-CHI2, TF-IG, and TF-RF. Meanwhile, the ANOVA and paired-sample *t*-test were implemented to test the significance of the differences in classification performance of different schemes. Moreover, the similarity analysis of term (feature) weighting is conducted. The results of a thorough experimental analysis clearly show that the proposed schemes TF-IEF, RTF-IEF, TF-RIEF, and RTF-RIEF outperform other term weighting schemes such as TF-IDF, TF-CHI2, TF-IG, and TF-RF. In addition, they can achieve the better classification performance and quite robust regardless of the corpus. Specifically, one of the improved versions of TF-IEF, RTF-RIEF, performs best in almost all the cases.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under grant 51435011, by the Science & Technology Ministry Innovation Method Program under grant 2017IM040100, and by the Sichuan Applied Foundation Project under grant 2018JY0119.

ORCID

Zhong Tang https://orcid.org/0000-0003-0949-3692
Wenqiang Li https://orcid.org/0000-0002-9592-4454

REFERENCES

1. Uysal AK, Gunal S. A novel probabilistic feature selection method for text classification. *Knowl Based Syst*. 2012;36:226-235.
2. Chen K, Zhang Z, Long J, Zhang H. Turning from TF-IDF to TF-IGM for term weighting in text classification. *Expert Syst Appl*. 2016;66(33):245-260.
3. Canuto S, Sousa DX, Gonçalves MA, Rosa TC. A thorough evaluation of distance-based meta-features for automated text classification. *IEEE Trans Knowl Data Eng*. 2018;30(12):2242-2256.
4. Do T-N, Poulet F. Latent-ISVM classification of very high-dimensional and large-scale multi-class datasets. *Concurrency Computat Pract Exper*. 2019;31(2):e4224.
5. Gao L, Zhou S, Guan J. Effectively classifying short texts by structured sparse representation with dictionary filtering. *Information Sciences*. 2015;323(1):130-142.
6. Uysal AK, Gunal S. The impact of preprocessing on text classification. *Inf Process Manag*. 2014;50(1):104-112.
7. Ay Karakuş B, Talo M, Hallaç İR, Aydin G. Evaluating deep learning models for sentiment classification. *Concurrency Computat Pract Exper*. 2018;30(21):e4783.
8. Lee J-H, Yeh W-C, Chuang M-C. Web page classification based on a simplified swarm optimization. *Appl Math Comput*. 2015;270(1):13-24.
9. Sabbah T, Selamat A, Selamat MH, Ibrahim R, Fujita H. Hybridized term-weighting method for dark web classification. *Neurocomputing*. 2016;173(15):1908-1926.
10. Li C, Liu S. A comparative study of the class imbalance problem in Twitter spam detection. *Concurrency Computat Pract Exper*. 2018;30(5):e4281.
11. Méndez JR, Cotos-Yañez TR, Ruano-Ordás D. A new semantic-based feature selection method for spam filtering. *Appl Soft Comput*. 2019;76:89-104.
12. Stamatatos E. Author identification: using text sampling to handle the class imbalance problem. *Inf Process Manag*. 2008;44(2):790-799.
13. Li JS, Chen L-C, Monaco JV, Singh P, Tappert CC. A comparison of classifiers and features for authorship authentication of social networking messages. *Concurrency Computat Pract Exper*. 2017;29(14):e3918.
14. Coussement K, Van den Poel D. Integrating the voice of customers through call center emails into a decision support system for churn prediction. *Inf Manag*. 2008;45(3):164-174.
15. Sebastiani F. Machine learning in automated text categorization. *ACM Comput Surv*. 2002;34(1):1-47.
16. Zhang W, Yoshida T, Tang X. A comparative study of TF*IDF, LSI and multi-words for text classification. *Expert Syst Appl*. 2011;38(3):2758-2765.
17. Li Z, Xiong Z, Zhang Y, Liu C, Li K. Fast text categorization using concise semantic analysis. *Pattern Recognit Lett*. 2011;32(3):441-448.
18. Wang D, Zhang H. Inverse-category-frequency based supervised term weighting schemes for text categorization. *J Inf Sci Eng*. 2013;29(2):209-225.
19. Melucci M. Vector-space model. In: Liu L, Özsu MT, eds. *Encyclopedia of Database Systems*. Boston, MA: Springer US; 2009.
20. Lan M, Tan CL, Su J, Lu Y. Supervised and traditional term weighting methods for automatic text categorization. *IEEE Trans Pattern Anal Mach Intell*. 2009;31(4):721-735.
21. Sabbah T, Selamat A, Selamat MH, et al. Modified frequency-based term weighting schemes for text classification. *Appl Soft Comput*. 2017;58:193-206.
22. Agnihotri D, Verma K, Tripathi P. Variable global feature selection scheme for automatic classification of text documents. *Expert Syst Appl*. 2017;81(15):268-281.
23. Xia R, Zong C, Li S. Ensemble of feature sets and classification algorithms for sentiment classification. *Information Sciences*. 2011;181(6):1138-1152.
24. Altınel B, Ganiz MC. Semantic text classification: a survey of past and recent advances. *Inf Process Manag*. 2018;54(6):1129-1153.
25. Rao G, Huang W, Feng Z, Cong Q. LSTM with sentence representations for document-level sentiment classification. *Neurocomputing*. 2018;308:49-57.

26. Nguyen HT, Duong PH, Cambria E. Learning short-text semantic similarity with word embeddings and external knowledge sources. *Knowl Based Syst*. 2019;182. https://doi.org/10.1016/j.knosys.2019.07.013
27. Kastrati Z, Imran AS, Yayilgan SY. The impact of deep learning on document classification using semantically rich representations. *Inf Process Manag*. 2019;56(5):1618-1632.
28. Zhang D, Xu H, Su Z, Xu Y. Chinese comments sentiment classification based on word2vec and SVM^{perf}. *Expert Syst Appl*. 2015;42(4):1857-1863.
29. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. Paper presented at: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems; 2013; Lake Tahoe, NV.
30. Harris ZS. Distributional structure. *Word*. 1954;10(2-3):146-162.
31. Kim HK, Kim H, Cho S. Bag-of-concepts: comprehending document representation through clustering words in distributed representation. *Neurocomputing*. 2017;266:336-352.
32. Le QV, Mikolov T. Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on Machine Learning; 2014; Beijing, China.
33. Kim D, Seo D, Cho S, Kang P. Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec. *Information Sciences*. 2018;477:15-29.
34. Maisonnave M, Delbianco F, Tohmé F, Maguitman A. A flexible supervised term-weighting technique and its application to variable extraction and information retrieval. *Inteligencia Artificial*. 2019;22(63):61-80.
35. Sinoara RA, Camacho-Collados J, Rossi RG, Navigli R, Rezende SO. Knowledge-enhanced document embeddings for text classification. *Knowl Based Syst*. 2019;163(1):955-971.
36. Zheng HT, Wang Z, Wang W, Sangaiah AK, Xiao X, Zhao C. Learning-based topic detection using multiple features. *Concurrency Computat Pract Exper*. 2018;30:e4444.
37. Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. *Inf Process Manag*. 1988;24(5):513-523.
38. Dogan T, Uysal AK. Improved inverse gravity moment term weighting for text classification. *Expert Syst Appl*. 2019;130:45-59.
39. Pinto D, Gómez-Adorno H, Vilariño D, Singh VK. A graph-based multi-level linguistic representation for document understanding. *Pattern Recognit Lett*. 2014;41:93-102.
40. Kim S-B, Han K-S, Rim H-C, Myaeng SH. Some effective techniques for naive Bayes text classification. *IEEE Trans Knowl Data Eng*. 2006;18(11):1457-1466.
41. Salton G, Wong A, Yang CS. A vector space model for automatic indexing. *Commun ACM*. 1974;18(11):613-620.
42. Jones KS. A statistical interpretation of term specificity and its application in retrieval. *J Doc*. 1972;28(1):11-21.
43. Debole F, Sebastiani F. Supervised term weighting for automated text categorization. In: Proceedings of the ACM Symposium on Applied Computing; 2003; Melbourne, FL.
44. Altınçay H, Erenel Z. Analytical evaluation of term weighting schemes for text categorization. *Pattern Recognit Lett*. 2010;31(11):1310-1323.
45. Ren F, Sohrab MG. Class-indexing-based term weighting for automatic text classification. *Information Sciences*. 2013;236(1):109-125.
46. Liu Y, Loh HT, Sun A. Imbalanced text classification: a term weighting approach. *Expert Syst Appl*. 2009;36(1):690-701.
47. Quinlan JR. Induction of decision trees. *Machine Learning*. 1986;1(1):81-106.
48. Erenel Z, Altınçay H. Nonlinear transformation of term frequencies for term weighting in text categorization. *Eng Appl Artif Intell*. 2012;25(7):1505-1514.
49. Dogan T, Uysal AK. On term frequency factor in supervised term weighting schemes for text classification. *Arab J Sci Eng*. 2019. https://doi.org/10.1007/s13369-019-03920-9
50. Porter MF. An algorithm for suffix stripping. *Program*. 2006;40(3):211-218.
51. Şahin DÖ, Kılıç E. Two new feature selection metrics for text classification. *Automatika*. 2019;60(2):162-171.
52. Yang Y, Pedersen JO. A comparative study on feature selection in text categorization. In: Proceedings of the 14th International Conference on Machine Learning; 1997; Nashville, TN.
53. Meng J, Lin H, Yu Y. A two-stage feature selection method for text categorization. *Comput Math Appl*. 2011;62(7):2793-2800.
54. Wang S, Pedrycz W, Zhu Q, Zhu W. Subspace learning for unsupervised feature selection via matrix factorization. *Pattern Recognition*. 2015;48(1):10-19.
55. Liu H, Yu L. Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans Knowl Data Eng*. 2005;17(4):491-502.
56. Taşcı Ş, Güngör T. Comparison of text feature selection policies and using an adaptive framework. *Expert Syst Appl*. 2013;40(12):4871-4886.
57. Zong W, Wu F, Chu LK, Sculli D. A discriminative and semantic feature selection method for text categorization. *Int J Prod Econ*. 2015;165:215-222.
58. Forman G. An extensive empirical study of feature selection metrics for text classification. *J Mach Learn Res*. 2003;3:1289-1305.
59. Quan X, Liu W, Qiu B. Term weighting schemes for question categorization. *IEEE Trans Pattern Anal Mach Intell*. 2011;33(5):1009-1021.
60. Farid DM, Zhang L, Rahman CM, Hossain MA, Strachan R. Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks. *Expert Syst Appl*. 2014;41(4):1937-1946.
61. Vapnik VN. *The Nature of Statistical Learning Theory*. New York, NY: Springer; 1995.
62. Yang YM. An evaluation of statistical approaches to text categorization. *Information Retrieval*. 1999;1(1-2):69-90.

How to cite this article: Tang Z, Li W, Li Y. An improved term weighting scheme for text classification. *Concurrency Computat Pract Exper*. 2019;e5604. https://doi.org/10.1002/cpe.5604